REMARKS

The Examiner has rejected claims 1-9, 11-15 and 17 under 35 U.S.C. §102 (b) as being anticipated by US patent number 5,758,079 (Ludwig). The Examiner contends that Ludwig teaches a user interface and a plurality of application wrappers that provide an interface between the global operating system and each of a plurality of software applications.

The Ludwig system does not teach applicants system in that it is a peer to server network rather than a peer to peer as is the case for applicant's invention. In the embodiments described in the '079 patent all applications and software modules are loaded on each individual collaborative multimedia workstation (CMW) and standard function calls and interapplication protocols are used. Because of this it becomes clear that many applications, such as for example Autocad®, cannot be worked on simultaneously because Autocad® is not designed to permit simultaneous editing. Ludwig permits a snapshot to be distributed to session participants but only one CMW can actually control the data and edit the drawing. Other participants can make annotations which would be communicated on their snapshot back to the originating CMW.

Ludwig further requires that a collaboration initiator be incorporated in the CMW of each computer in the session. The collaboration initiator retrieves the necessary address information and invokes the necessary applications using standard function calls and standard inter-application protocols. It will also query a service server module for address information of the other session participants and coordinate session setup with the corresponding collaboration initiators on the participating CMWs.

The present invention relates to a real-time collaboration which is a multi-computer operator application that allows computer users in a collaborative session to <u>exchange and actively use application data in real-time</u>. Users at remote locations can both work in Autocad® on the same drawing essentially simultaneously, which is normally not readily accomplished and also share data in real time in Exel® or Word®. Likewise the present invention permits multiple players in a game to play the game in real-time while simultaneously seeing and talking to one another by audio/video. The present invention allows simultaneous sharing of data in programs that normally do not allow more than one user to manipulate data in the program. The present invention provides a user with a simple and straightforward way to enter into a complex multi-application collaborative session while simultaneously linking shared data and shared streaming broadband data needs.

The ability to have multiple collaborative applications function in real-time for linking shared project data and shared broadband data needs is provided by the <u>application wrapper</u>. The application wrapper cooperates with the operating system and with the Real-Time Knowledge Center to cause dissimilar applications to function together in the collaborative session. As described in the specification at page 9, beginning line at line 26, the application wrapper can be analogized to a printer driver. Thus the application wrapper is a software module that communicates with the operating system to control a particular application and execute that application or a specific file function for operation with dissimilar applications or to modify a program for real time data sharing between session computers. Each session computer includes the same file wrapper that provides executable instructions so that the session application or the same file operations are run on each session computer. In the case of certain applications, like AutoCAD®, which normally will not allow real-time data sharing of the same drawing file

between session computers, the application wrapper, in cooperation with the Real-Time Knowledge Center, substitutes a shell for the normal shell of the program to allow the operating system to execute the program for a real-time data sharing. As described the application wrapper comprises several parts, a first part that provides an application-specific computer operator interface and a second part that controls a corresponding collaborative application. In addition the application wrapper comprises a keyboard filter so that key board functions between applications in a session are the same. The application wrapper thus in effect contains the shell that communicates with the kernel of the operating system for execution of different applications and/or to provide desired functions in different applications operating on session computers in a collaborative session.

Is submitted that the Examiner has failed to make a prima facie case for anticipation. The Federal Circuit has stated that "anticipation requires the disclosure in a single prior art reference of each element of the claims under consideration" *W.L. Gore & Associates v. Garlock,* 721 F. 2nd 1540, 220 USPQ 303 (Fed. Cir., 1983) *cert. denied,* 469 U.S. 851 (1984) further, "under 35 USC § 102, anticipation requires that the prior art reference must be enabling, thus placing the allegedly disclosed matter in the possession of the public. *Akzo N.V. v. U.S. Int'l Trade Comm'n,* 808 Fed. 2nd 1471, 1USPQ 2d 1001, 1010 (Fed. Cir. 1991).

Ludwig fails to disclose the application wrapper module of applicant's invention and it follows that there is no teaching or enablement of the application wrapper feature. Ludwig uses software modules to communicate with the operating system/GUI software and other applications utilizing standard function calls and interapplication protocols. Ludwig describes a "collaboration initiator" that does not function as, nor is it the equivalent to, the application wrapper of applicants invention. It is clear from the above quoted statement that the concept of

the application wrapper was not recognized by Ludwig. As mentioned above the application wrapper of applicant's invention provides unique functions that permit in an operator friendly manner real time sharing of data and simultaneous operation of dissimilar applications. Clearly, application wrappers are not present in the Ludwig's system and are not disclosed by the reference.

In addition Ludwig requires multimedia servers, or their equivalent, in the network to provide, in conjunction with audio video switching devices, real-time audio and video storage servers for file management and for support of real-time audio/video recording and playback. In this connection the audio/video server of Ludwig provides a real-time transmission path and appropriate audio/video network ports for each computer accessing the server. This serves as yet another distinction of the present invention over Ludwig. Applicant's invention describes a network that is peer to peer, particularly for broadband streaming, while Ludwig is peer to server. Depending on the number of computers in the session, Ludwig's server can require a plurality of high bandwidth streaming application data channels that can present a bottleneck to audio/video and data transfer.

The Examiner has rejected claims 2 and 13 as anticipated by Ludwig. The Examiner states that it is inherent that the user login feature of Ludwig provide at least one security password for each of said collaborative sessions and that the passwords be stored in a centralized data index. Although most likely this would be the normal course of events, depending on the type of network, one may not use passwords but rely on the computer addresses that are stored in the Real Time Knowledge Center. For the sake of argument,however, we will accept the Examiner's statement. However, claims 2 and 13 are allowable over Ludwig since they both depend from independent claims which include the novel and unique application wrapper. As

stated above Ludwig did not contemplate the application wrapper and because claim 2 and claim 13 contain all the limitations of claim 1 and claim 11, respectively, Ludwig cannot anticipate either claim because an essential element of the claimed invention is not present.

Claims 3 and 12 have been rejected as anticipated by Ludwig. These claims call for the provision of network addresses for each client computer system. As pointed out above in connection with claims 2 and 13, claims 3 and 12 depend from claim 1 and claim 11 respectively and as such contain all the limitations of those claims including the application wrapper which is not disclosed by Ludwig. Accordingly, Ludwig cannot be properly used to reject these claims under § 102.

Examiner has rejected claims 4 and 14 as anticipated by Ludwig. Claim 4 has been amended to include the step of loading a corresponding application wrapper, which step is clearly not disclosed in Ludwig. Claim 14 has been amended to include executable application instructions embedded in an application wrapper. In addition, as pointed out above in connection with claims 2, 3, 11 and 12, Ludwig does not contemplate the application wrapper modules as called for in claims 1 and 11 from which the foregoing claims 2, 3, 4 and 12, 13 and 14 depend respectively.

With regard to claim 5, which also depends from claim 1, it is clear that Ludwig does not anticipate this claim since any user joining or leaving a collaborative session must have corresponding application wrappers in order to simultaneously run the applications called for by the session. Since Ludwig does not teach the use of application wrappers there can be no anticipation with regard to this claim.

Regarding claim 6, the portion of Ludwig relied on by the Examiner refers to snapshot sharing which permits the other participants to telepoint or annotate the snapshot. This can

hardly be said to comprise real-time data sharing between applications since only the originator can modify the graphic or other data. In applicant's invention the graphic or other data can be modified or edited by all of the session users. Such real-time data sharing is achieved by the application wrappers as defined in the instant application that allow each session computer to operate the particular application involved and to access and modify the application data in real-time. With regard to a plurality of data channels it should be noted that Ludwig runs at least the audio video portions through a multimedia server while in the present invention the data channels, including audio/video, are opened directly between session user computers. Thus Ludwig defines a peer to server network while applicant defines a peer to peer network. In addition, claim 6 depends from claim 1 which calls out application wrappers and as a dependent claim, claim 6 contains all of the limitations of claim 1. Ludwig does not disclose nor contemplate an application wrapper or its function.

Claim 8 has been amended to even more clearly point out that the data streaming channels are formed directly between the computers in the session. This is to be distinguished from Ludwig whose channels, particularly audio/video channels extend between a computer, a multi media server and finally to another computer or computers in the session.

Claim 9 has been rejected as anticipated in view of the description by Ludwig of the appointment of an expert as the collaboration administrator. Claim 9 depends from claim 1 and therefore contains all the limitations, particularly the application wrapper and its unique functions of claim 1. Thus an administrator in a session running the Ludwig network would not the equivalent of an administrator running the system of the present invention. Ludwig does not contemplate the use of application wrappers to permit simultaneous data sharing between remote

computers and the simultaneous operation of applications and concurrent operation of dissimilar applications.

The Examiner states with regard to claim 15 that Ludwig teaches a real-time knowledge Center which downloads or runs remotely any required applications. The portion of Ludwig cited by the Examiner (col. 23, line 59 to col. 24, line 4) appears to relate only to the fact that the several local area networks in the Ludwig system are able to communicate with one another. Ludwig does teach that the individual workstations contain the standard multitasking operating system/GUI software as well as other applications such as word processing and spreadsheet programs. Software modules are provided to communicate with the operating system/GUI software and other applications using standard function calls an interapplication protocols (col. 18, line 41 – 49). With applicant's system, however, a user may download application wrappers from a remote server that will permit him to work simultaneously with another remote computer using programs that are not designed to work together in that manner. Real-time data sharing is accomplished with applicants system as contrasted with "snapshot" data sharing which can hardly be said to comprise real-time data sharing since the recipients of the snapshots can only make annotations but cannot actually access and edit the data. As mentioned above the best example of this would be to remote computer's operating on the same drawing in Autocad® with both computers being able to simultaneously edit the common Autocad® data.

The indication of allowable subject matter in claims 10, 16 and 18 is noted with appreciation. In this regard claims 10, 16 and 18 have been canceled and the language of those claims has been included in claims 11 and 17. In this manner it can be considered that these claims now exist in independent form and are now deemed to be allowable.

It should be pointed out, however, that the keyboard filter as now defined in claims 19, 20 and 21 is in fact a part of an application wrapper and is a function provided by an application wrapper among other functions needed to run dissimilar applications without conflict. Quoting from the specification, page 10 beginning at line 8,:

> The application wrapper 24 in the global operating system 20 has known interfaces used by the global operating system 20. The application rapper high as peculiarities of interfacing in controlling a given application from the global operating system 20 therefore, when a new application is used within the global operating system 20, it does not require changes in the global operating system 20 client software. Instead a novel software module, the 'the application wrapper' 24 is produced. It is comprised of a first part that provides an application-specific computer operator interface, a second part that controls a corresponding collaborative application, and a keyboard filter, ...
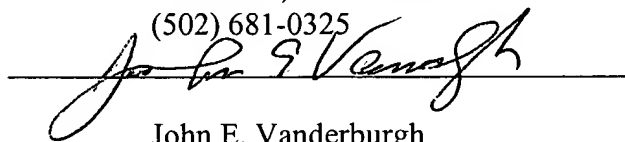
It follows that the keyword filter is a function provided by the application wrapper that will allow dissimilar applications to work together. Needless to say there are many other functions that must be programmed into an application wrapper module to control other features of dissimilar applications to permit them to work together without conflict.

In view of the foregoing amendments and remarks, the present application is clearly novel and patently distinct from the system defined by Ludwig. At least one central element is not disclosed by Ludwig and consequently it cannot support an anticipation rejection.

Accordingly, it is respectfully requested that the Examiner issue a notice of allowance.

Date: December 14, 2004   Respectfully Submitted,

                                                 STITES & HARBISON, PLLC
                                                 400 West Market Street, Suite 1800
                                                 Louisville, KY 40202-3352
                                                 (502) 681-0325

                                                 John E. Vanderburgh
                                                 Registration No. 24,041